DOCUMENT RESUME

ED 422 939 IR 057 097

AUTHOR	Sim, Edward
TITLE	Teaching OOA: Issues and Challenges.
PUB DATE	1997-00-00
NOTE	7p.; In: Proceedings of the International Academy for Information Management Annual Conference (12th, Atlanta, GA, December 12-14, 1997); see IR 057 067.
PUB TYPE	Reports - Evaluative (142) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Computer System Design; Course Content; Curriculum Development; Higher Education; Information Science Education; *Instructional Design; *Management Information Systems; Models; *Systems Analysis; *Systems Development; Teaching Methods
IDENTIFIERS	*Object Orientation; Object Oriented Programming

ABSTRACT

This paper presents the argument that object oriented analysis (OOA) is rapidly becoming an important systems analysis methodology and that current systems analysis and design courses should present OOA. However, because of the embryonic nature and rapidly changing content of OOA, instructors are faced with special challenges when designing OOA curriculum. This paper, based on research and teaching in this area, identifies several of these issues and challenges and makes recommendations and suggestions for meeting these challenges. Specifically the following issues are addressed: how much of the content of the class should be devoted to OOA; how the sequence of material should be organized; and whether one or more specific methodologies should be taught. The challenges of teaching these new concepts are considered, and limitations posed by conflicting terminologies and notation are reviewed. (Contains 26 references.) (Author/AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

TEACHING OOA: ISSUES AND CHALLENGES

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

T. Case

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Edward Sim
Loyola College

ED 422 939

This paper presents the argument that object analysis (OOA) is rapidly becoming an important systems analysis methodology and that current systems analysis and design courses should present OOA. However, because of the embryonic nature and rapidly changing content of OOA, instructors are faced with special challenges when designing OOA curriculum. This paper, based on our research and teaching in this area identifes several of these issues and challenges and makes recommendations and suggestions for meeting these challenges. Specifically we address the issues of: How much of the content of class should be devoted to OOA; How the sequence of material should be organized; and whether one or more specific methodologies should be taught. Further, we address the challenges of teaching these new concepts and review the limitations posed by conflicting terminologies and notation.

INTRODUCTION

This paper will report on our effort to teach objected oriented analysis techniques and methodologies in our graduate level MIS system design course. Since many universities are currently seeking to include these object oriented (OO) concepts and methodologies in their curriculums, we believe that it is important to share our experiences with teaching OO methodologies in our system design course. The increasing use of OO concepts in all aspects of IS and IT has made knowledge of these concepts imperative for the IS professional (Pancake, 1995). Recent literature suggests that OO concepts, techniques, and methodologies will play an increased role in the design, development and implementation of organizational ISs and that analysts, designers and others should have familiarity with these concepts (Lewis, 1996; Vessey & Conger, 1994a; Vessey & Conger, 1994b). This paper will address some of the major issues and challenges presented in designing a system design course based on OO concepts and will address some of the pedagogical issues of teaching OO concepts. In addition, we plan to make recommendations concerning how these issues and challenges can be met.

BACKGROUND

The system design course taught at our university is a required course for our graduate MIS students. The course, which we teach at the 600 level, is expected to be taken by the students at the beginning of their second year of graduate study. However, students with IS backgrounds and/or system development experience may take the course earlier in their studies. Many of our students are working professionals who are taking the MIS program for career advancement within the IS field or for career transition to the IS field. Most of our students were business majors as undergraduates but a significant percentage were computer science majors. Although an understanding of a programming language is required for admission to our MIS graduate program, most of our students have minimal programming skills. Very few students have any knowledge of an object oriented programming language.

The primary focus of the system design course has been on analysis and logical design of information and software systems. Although the effect of physical implementation issues on system performance are addressed in the course, physical or low level design issues are not a primary focus of the course. Three years ago the system design course was taught strictly as a structured analysis and design course. At that point, the course was taught primarily as a structured methodology course. Scant attention was given to completing paradigms of development or to the conceptual foundations of the various development methodologies.

IR057097

However, since that time the system design course has gradually developed into a course with a significant emphasis on object-oriented concepts and methodologies. It is that experience on which the recommendations presented in this paper are based.

ISSUES

During this transition of our system design course, several key issues have arisen concerning the concepts and topics presented. In addition, certain challenges are inherently faced when new topics and concepts are introduced into any course. Once the decision has been made to include OO concepts in a system design course, several key 'content' issues must be addressed:

- How much of the course should be devoted to OO concepts ?
- Should the OO concepts and methodologies be interleaved with structured concepts or should the material be presented sequentially ?
- Should a specific OO methodology be presented in depth or should the OO methodologies be surveyed ?

Over the past three years we have taken a number of approaches to each of the above issues and consequently see some merit to each approach used.

AMOUNT OF TIME DEVOTED TO OO CONCEPTS

Given the current course content in most analysis and design courses taught in business schools, the addition of 'new' OO material presents complicated time management problems for the instructor designing the course. However, the need to maintain currency in the course and to prepare students for practices in industry require that this new material be added. Many textbook authors have recognized the need to include this material and have added OO material to their texts (Dewitz, 1996; Pressman, 1996; Whitten, Bentley & Barlow, 1994). However, the amount of time devoted to this new material should depend on the instructors perception of its value and the overall learning objectives of the course.

Our recommendation is that at least 30 % of the course be devoted to the OO concepts for courses that follow standard analysis and design formats. Devoting less time to this material often presents difficulty in covering most of the advanced concepts (i.e., object interaction diagrams, object life-cycles, etc.) that are needed to present a good conceptual overview of the capability and limitations of the OO approach. Given the allocation of four weeks in a typical analysis and design course the following sequence of topics could be covered in some detail.

Week 1: Introduction to OO concepts

Objects, Classes, Inheritance, Operations and Behavior, polymorphism, encapsulation

Overview of advantages and disadvantages of OO approach

Week 2: Modeling with OO Concepts

Introduction of OO models;

Structural Modeling (object models)

Dynamic and Behavioral Modeling (functional modeling, object interaction diagrams, use cases, object life cycles)

Week 3: OOAD Methodologies

Overview of OOAD methodologies

Coad & Yourdon (Coad & Yourdon, 1991a; Coad & Yourdon, 1991b)

Booch (Booch, 1994)

OMT (Rumbaugh, Blaha, Premerlani, Eddy & Lorensen, 1991)

Week 4: Case Study

Complete implementation of OOAD with a particular example (Coad, North & Mayfield, 1995)

Unless the OO analysis and design topics are covered in a separate 'advanced' class we do not recommend that less than four weeks be spend on the above topics. The 'paradigm' shift of OO is

often difficult for students to absorb in less time. The disadvantage of using more time for coverage of these concepts is that often other important topics are often neglected. Given that most of the industry is still using the traditional structured approach to systems analysis and design, we feel that it is important that these topics be covered as well.

PRESENTATION OF MATERIAL

Basically we have found that organizing the OO sequence of material into a separate topic has worked better for us than interleaving the material throughout the course. However, we can see some important advantages with a sequencing format that interleaves OO material with the structured approach. Dewitz in her recent textbook: System Analysis and Design and the Transition to Objects does a good job of integrating OO concepts with structured analysis and design concepts. The advantage of this approach is that similar examples can be used with both approaches and that the similarities between the approaches can be illustrated within the same context. For example, the concepts and notation of an entity-relationship diagram can be used to introduce the basic object model which shows structural relationships between system objects. However, the disadvantage of this approach is that it fails to address the necessary paradigm shift of OO. Students exposed to this presentation sequence often fail to appreciate the real power of OO techniques.

SURVEY OF METHODOLOGIES OR JUST ONE ?

OO methodologies are still relatively new and consequently many different OO approaches have been proposed and developed (de Champeaux & Fare, 1992; Eckert & Golder, 1994; Hutt, 1994). A recent survey of the OO literature identified over 27 different OOAD methodologies (Strouse, 1995). Despite the effort of OMG to promote some standards for OOAD, new methodologies base on different concepts and notation are continuing to be developed. Given the chaotic nature and rapid development of OOAD, it may be a disservice to students to focus exclusively on just one methodology. Our recommendation would be to survey several of the more prominent OOA methodologies. In our course we typically use the Rumbaugh's OMT, Booch, and Odell & Martin (Booch, 1994; Martin & Odell, 1992; Rumbaugh et al., 1991). However, we have also used Embley, Coad & Yourdon and Jacobson methodologies (Coad & Yourdon, 1991a; Coad & Yourdon, 1991b; Embley, Kurtz & Woodfield, 1992; Jacobson, 1993). Our general impression is that the students typically find the Coad & Yourdon methodology easiest to understand; however, we feel that its informality may limit its adoption in industry.

The disadvantage of surveying several approaches vis a vis a complete treatment of one approach is handling the different notation and jargon. Students can easily get confused with the different terms and models associated with each approach. Our approach generally has been to introduce the basic modeling concepts in a relatively methodologically independent way and then to show the various implementations of the basic concept being modeled. For example, modeling structure is essential in analysis and design and most OOA approaches use some form of an 'object model' to show most aspects of system structure. Specifically, we might show an entity relationship diagram as a example of a structural model and then show how an object model uses similar concepts to model structure. Typically we would start with a relatively easy model notation like Coad and Yourdon's and then move to a more complicated and robust notation like Embley's.

OO CHALLENGES

Despite the widespread interest in OO concepts and relatively recent publication of many new OO methodologies, many of the OO concepts as they are applied to analysis and high level design are relatively informal and fluid (Embley, Jackson & Woodfield, 1995; Hutt, 1994). Compounding this problem is the confusing and sometimes conflicting terminology that has emerged from the OO methodology competition (Monarchi & Puhr, 1992). This confusion can be frustrating to an instructor seeking to organize class material that presents the OO concepts in a relatively straightforward way. Teachers who want to teach OO concepts for analysis and design should be aware of the following pedagogical issues:

- Although the basic concepts of object orientation are relatively agreed on, the implementation of those ideas for analysis and

design differs between OO methodologies (Synder, 1993).

- The modeling of system behavior within the OO paradigm is more problematic than the modeling of system structure (Iivari, 1995).
- OO methodologies generally require expertise in several modeling techniques (i.e.., use case, object life cycles, object interaction diagrams, etc.) that may differ between methodologies (Yourdon, Whitehead, Thomann, Oppel & Neverman, 1995).

DIFFERENT IMPLEMENTATIONS

As mention earlier, the use of OO concepts for the analysis and design phases of system design are relatively new and are still under development. Unlike object oriented concepts in object oriented programming languages, the application of the OO concepts to analysis and design is usually problematic and can be interpreted in numerous ways; consequently, different authors have implemented the OO concepts in different ways in their methodologies. Thus in many cases, the same terminology is used to represent different concepts and different terminologies are used to represent the same concepts. New terms are often introduced for old concepts in an effort to spruce up old ideas and create a proprietary vocabulary. Even the notational symbols used in the various methodologies are used to enhance the distinctiveness of the methodology and give proprietary rights to the progenitor; consequently, similar symbols (and terms) are often used to represent different concepts and different symbols are used to represent the same concepts (Strouse, 1995). However, instructors should not be too discouraged by this type of confusion since it has historically effected most disciplines. (Even the 'stately' mathematics had its notional wars such as the difference of calculus notation between Newton and Leibniz). It does require, however, that instructors be diligent and 'keep up' with the ever-changing vocabulary of OOA. In our course we have tried to keep the jargon to a minimum and to tailor our dialect to the background of the students.

MODELING SYSTEM BEHAVIOR

A recurrent criticism of most OOA methodologies is that they are poor at modeling system behavior (Fichman & Kemerer, 1992; Iivari, 1995). Critics have argued that most OOA approaches are poor at providing the necessary models to completely describe system functionality and behavior. In most cases, OOA methodologies use some form of state transition diagrams (STD) to model system behavior (e.g., the dynamic model in OMT (Rumbaugh et al., 1991)). Although STDs can be used to effectively model system behavior, it has been our experience that functional and behavioral modeling represent the hardest concepts for students to understand and model. It is particularly difficult for students to integrate behavior or functionality that may be modeled with a STD back to the assignment of a specific operation to an individual object. In addition we have found that some students have difficulty modeling the interaction of object behaviors necessary to achieve a specific system level function. Our basic approach to teaching these concepts is to thoroughly cover the STD model and to examine the particular approaches used by specific OOA methodologies only after the student completely understands the basic concepts of events, states, transitions. etc. We have found that Martin and Odell's Object Behavior Analysis (OBA) (Martin & Odell, 1992) and Jacobsons's use cases to be good starting points for illustrating behavioral modeling (Jacobson, 1993; Jacobson, 1995). OMT's event traces are effective for a good introduction into object interaction diagrams (Rumbaugh et al., 1991).

MULTIPLE MODELS

All OOA methodologies require the creation of one or more models to specify the requirements for a given system. Depending on the specific system and methodology used, one or more of the following models may be needed: object model, use cases, event trances, object interaction diagrams, functional model, event schema diagram, object life cycle. In many cases, presenting too many models can be confusing to the students. We try to limit our in-depth treatment to a few models and then illustrate the more advanced concepts with extensions to the existing models. We have found use cases, object interaction diagrams and object life cycle models to be particularly useful and understandable to the students. We emphasize that in most cases the information in these models can be completely integrated into the object model in

order to obtain a complete perspective of the system being modeled.

CONCLUSION

This paper has addressed some of the major issues and challenges of teaching OOA within the context of a system analysis and design course. As mentioned previously in our paper, OOA is a relatively new approach to system development and no clear dominant OOA methodology has emerged. Consequently, the ideas, terminology and notation of most OOA methodologies are changing rapidly. However, despite OOA shortcomings and limitations, many industry observers and proponents of OOA believe that OOA will eventually be as dominate as structured techniques. Thus we make the argument that object oriented methodologies and OO concepts in general should be major components of the basic analysis and design course. The difficulties of teaching OO in many instances are related to the newness of the approach and required paradigm shift. OOA, like most other things in the IS field, is changing rapidly. Consequently, this poses special challenges to the instructor wishing to incorporate these ideas into the analysis and design course. We hope that the suggestions and recommendations presented in this paper will be useful to all instructors who decide to teach OOA.

The recommendations and suggestions made in this article are based on our four years of experience of doing research and teaching object oriented analysis and design. The amount of empirical research in this field is very limited at this point and we are currently conducting several studies in this area. Future research needs to be done to validate the effectiveness of OOA and to evaluate the effectiveness of techniques for teaching OOA.

REFERENCES

Booch, G. (1994). Object-Oriented Analysis and Design with Applications. (2nd ed.). Redwood City, Calf.: Benjamin/Cummings Inc.

Coad, P., North, D., & Mayfield, M. (1995). Object Models: Strategies, Patterns, & Applications: Yourdon Press.

Coad, P., & Yourdon, E. (1991a). *Object-Oriented Analysis*. Englewood Cliffs, NJ: Yourdon Press.

Coad, P., & Yourdon, E. (1991b). *Object-Oriented Design*: Prentice Hall.

de Champeaux, D., & Fare, P. (1992). A comparative study of object-oriented analysis methods. *Journal of Object-Oriented Programming* (March/April), 21-33.

Dewitz, S. (1996). *System Analysis and Design and the Transition to Objects*: McGraw-Hill.

Eckert, G., & Golder, P. (1994). Improving object-oriented analysis. *Information and Software Technology*, 36(2), 67-85.

Embley, D., Jackson, R., & Woodfield, S. (1995). OO Systems Analysis: Is it or Isn't it ? *IEEE Software*, July, pp. 19-32.

Embley, D., Kurtz, B., & Woodfield, S. (1992). *Object-Oreinted Systems Analysis: A Model-Driven Approach*: Prentice Hall.

Fichman, R., & Kemerer, C. (1992). Object-Oriented and Conventional Analysis and Design Methodologies. *Computer* (October), 23-39.

Hutt, A. (Ed.). (1994). *Object Analysis and Design: Description of Methods*. New York: Wiley-QED.

Iivari, J. (1995). Object-Orientation as structural, functional and behavioral modeling: a comparsion of six methods for object-oriented analysis. *Information and Software Technology*, 37(3), 155-163.

Jacobson, I. (1993). *Object-Oriented Software Engineering*: Addison-Wesley.

Jacobson, I. (1995). *The Object Advantage: Business Process Reengineering with Object-Technology*: Addison-Wesley.

Lewis, M. (1996,). Getting the Requirements Right. *Object Magazine*, 42-47.

Martin, J., & Odell, J. (1992). *Object-Oriented Analysis and Design*. Englewood, NJ.: Prentice Hall.

Monarchi, D., & Puhr, G. (1992). A Research Typology for Object-Oriented Analysis and Design. *Communications of the ACM*, 35(9), 35-47.

Pancake, C. (1995). The Promise and Cost of Object Technology: A Five Year Forecast. *Communications of the ACM*, 33(10), 33-49.

Pressman, R. S. (1996). *Software Engineering: A Practitioner 's Approach*. (4th ed.). New York: McGraw-Hill, Inc.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). *Object-Oriented Modeling and Design*. Englewoods, NJ.: Prentice Hall.

Strouse, G. (1995). A Study of the Cognitive Transfer Efficiency in Analysis, Design, and Layering of Diagrammatic Notation Methodologies Employed in Object-Oriented Development. Unpublished Ph.d, University of Maryland, Baltimore County.

Synder, A. (1993). The Essense of Objects: Concepts and Terms. *IEEE Software* (January), 31-42.

Vessey, I., & Conger, S. (1994a). Learning to specify information requirements: the relationship between application and methodology. *Journal of Management Information Systems*, 10(2), 177-201.

Vessey, I., & Conger, S. (1994b). Requirements Specifications: Learning Object, Process, and Data Methodologies. *Communications of the ACM*, 37(5), 102-113.

Whitten, J., Bentley, L., & Barlow, V. (1994). *Systems Analysis and Design Methods*. (Third ed.). Homewood: Illinois: Irwin.

Yourdon, E., Whitehead, K., Thomann, J., Oppel, K., & Neverman, P. (1995). *Mainstream Objects: An Analysis and Design Approach for Business*. Upper Saddle River, NJ: Prentice Hall.